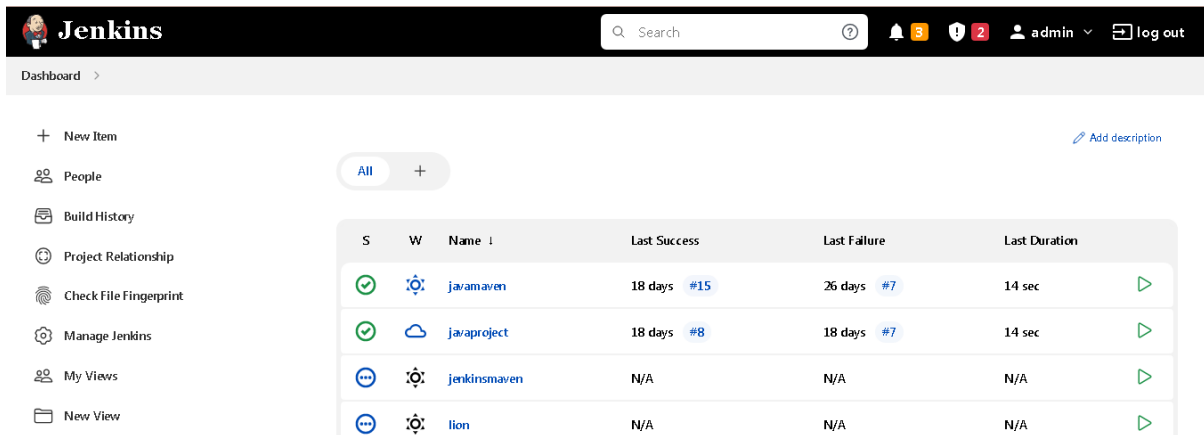


## Creating new jobs for java-maven project in Jenkins:

### 1. Click on dashboard in jenkins

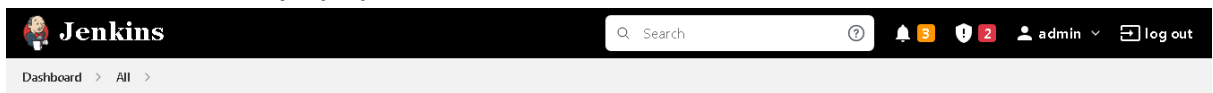


The screenshot shows the Jenkins Dashboard. The top navigation bar includes the Jenkins logo, a search bar, and user information (admin). The left sidebar lists various dashboard sections: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, and New View. The main content area displays a table of jobs with columns for Status (S), Workflow (W), Name, Last Success, Last Failure, and Last Duration. The table lists four jobs: 'javamaven', 'javaproject', 'jenkinsmaven', and 'lion'. The 'javamaven' and 'javaproject' jobs show successful builds, while 'jenkinsmaven' and 'lion' show no builds (N/A).

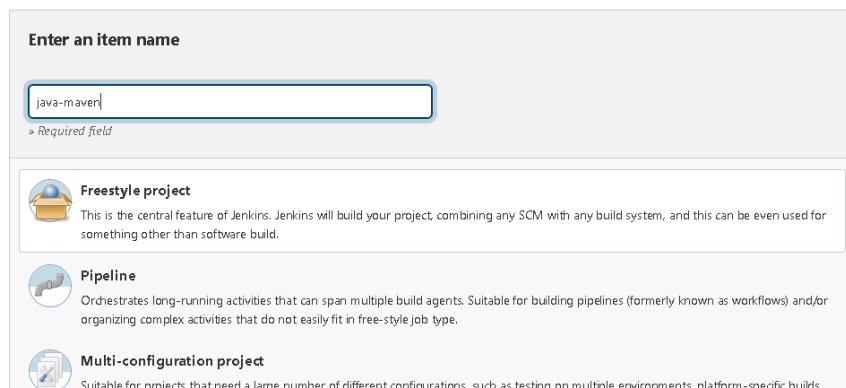
S	W	Name	Last Success	Last Failure	Last Duration
✓	🔧	javamaven	18 days #15	26 days #7	14 sec
✓	☁	javaproject	18 days #8	18 days #7	14 sec
...	🔧	jenkinsmaven	N/A	N/A	N/A
...	🔧	lion	N/A	N/A	N/A

### 2. Then click on new item

### 3. Then select freestyle project and enter item name, then click ok



The screenshot shows the Jenkins Dashboard with the breadcrumb 'Dashboard > All'. The top navigation bar is the same as the previous screenshot.



The screenshot shows the 'Enter an item name' form in Jenkins. The 'Item name' field contains 'java-maven'. Below the form, there are three options: 'Freestyle project', 'Pipeline', and 'Multi-configuration project'. The 'Freestyle project' option is selected.

**Enter an item name**

java-maven

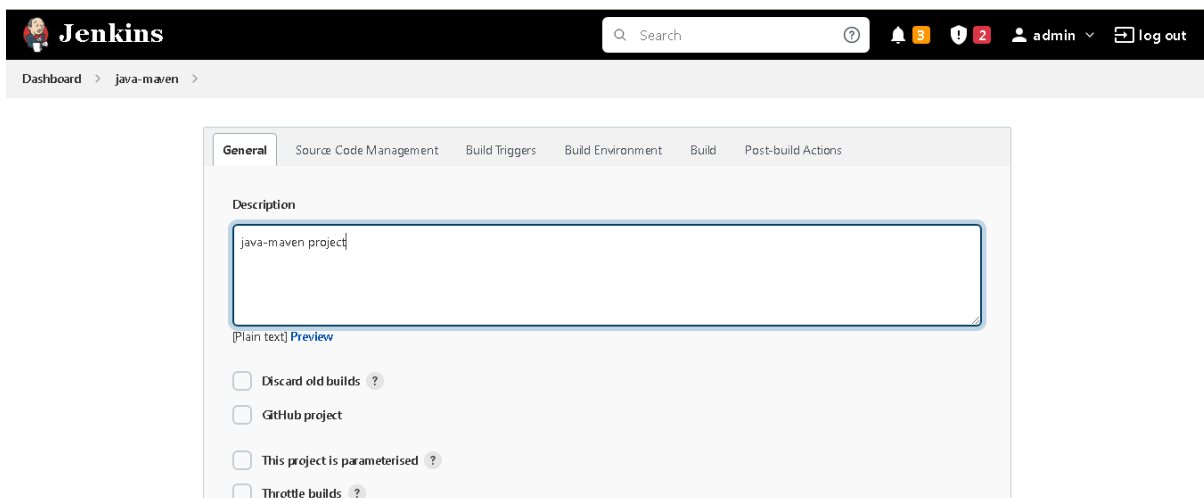
» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds.

### 4. In general tab, write description like java-maven project, then click on save



The screenshot shows the 'General' tab of the Jenkins job configuration page for 'java-maven'. The 'Description' field contains 'java-maven project'. Below the form, there are several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterised', and 'Throttle builds'. The 'Discard old builds' checkbox is checked.

**General** Source Code Management Build Triggers Build Environment Build Post-build Actions

**Description**

java-maven project

[Plain text] [Preview](#)

☒ Discard old builds ?

☐ GitHub project

☐ This project is parameterised ?

☐ Throttle builds ?

5. In source code management tab select git fill the repository url and credentials of github repository url, then click on save

Dashboard > java-maven >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/RAAGHAVENDRAA/javamaven.git

Credentials ?

raaghavend@gmail.com/\* (sure)

+ Add

Advanced...

Add Repository

Save Cancel

6. In source code management tab select branches to build an \*/master, then click on save

Dashboard > java-maven >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/master

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Save Cancel

7. In build tab select invoke top level maven target

Dashboard > java-maven >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Inspect build log for published Gradle build scans

☐ Terminate a build if it's stuck

☐ With Ant ?

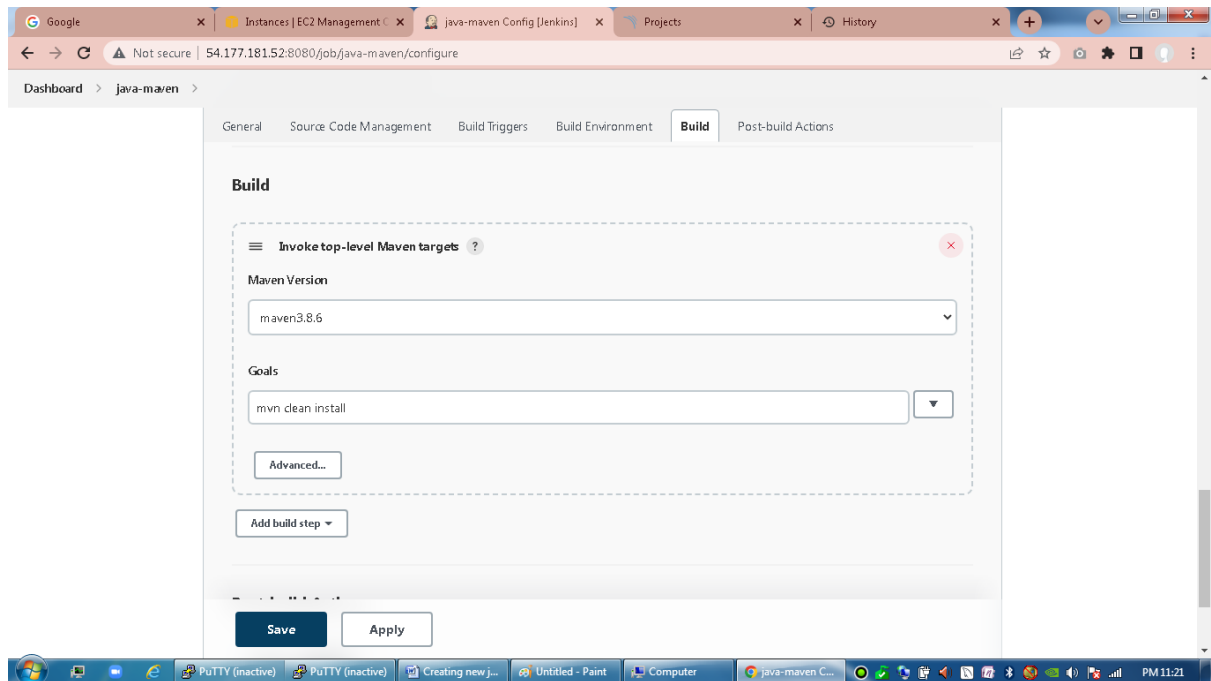
Build

Add build step ^

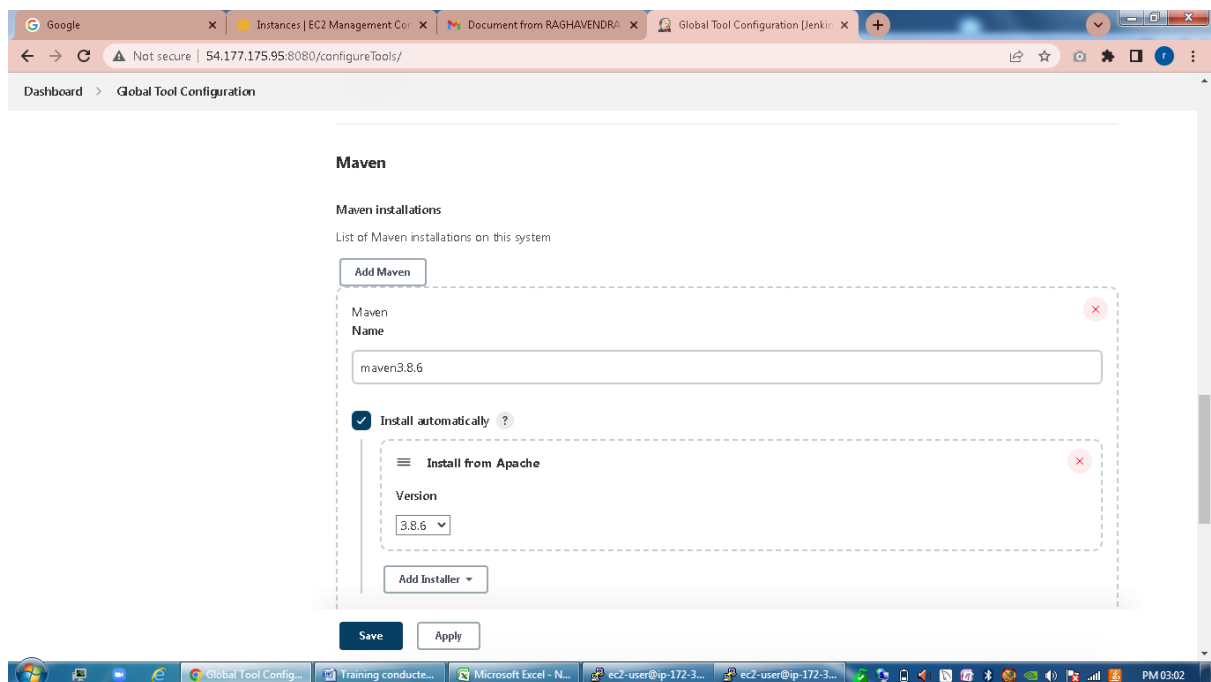
Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit


8. In build tab , select maven version maven3.8.6 and in goals put mvn clean install, then click on save





For maven installation goto global tool configuration. In maven installation click on add maven button do as shown in following figure



9. After save click on build now in left panel.

**Jenkins**

Search

 3 2adminlog out

Dashboard > java-maven >

↑ Back to Dashboard

Status

</> Changes

Workspace

▶ Build Now


⚙️ Configure


🗑️ Delete Project

✎️ Rename

## Project java-maven

java-maven project

 [Workspace](#)


 [Recent Changes](#)

### Permalinks



[Edit description](#)

Disable Project

## 10. Once click on build now we need to get following console output

**Jenkins**

Search

 3 2adminlog out

Dashboard > java-maven > #2

↑ Back to Project

Status

</> Changes

▶ Console Output

📄 View as plain text

⚙️ Edit Build Information

🔴 Git Build Data

← Previous Build

## Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/java-maven
The recommended git tool is: NONE
using credential sure
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/java-maven/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/RAAGHAVENDRAA/javamaven.git # timeout=10
Fetching upstream changes from https://github.com/RAAGHAVENDRAA/javamaven.git
> git --version # timeout=10
> git --version # 'git version 2.31.1'
using GIT_ASKPASS to set credentials sure
> git fetch --tags --force --progress -- https://github.com/RAAGHAVENDRAA/javamaven.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ demo ---
[INFO] Installing /var/lib/jenkins/workspace/java-maven/target/demo-0.0.1-SNAPSHOT.war to
/var/lib/jenkins/.m2/repository/com/example/demo/0.0.1-SNAPSHOT/demo-0.0.1-SNAPSHOT.war
[INFO] Installing /var/lib/jenkins/workspace/java-maven/pom.xml to /var/lib/jenkins/.m2/repository/com/example/demo/0.0.1-
SNAPSHOT/demo-0.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.662 s
[INFO] Finished at: 2022-09-21T18:03:24Z
[INFO] -----
Finished: SUCCESS
```